#### Example:1 Verify Docker Version

First important task while working Docker containers is to know your Docker version, Use below command to know your docker version

[root@docker ~]# docker --version

Docker version 17.03.1-ce, build c6d412e

[root@docker ~]#

**Example:2 View system wide Information**

‘**docker info**‘ command is used to view the system wide information like Docker’s root Directory, OS version, Kernel Version, Docker Version, RAM, CPU and Docker Registry.

**Example:3 Search Docker Images in its Registry Server**

With help of ‘**docker search**‘ command we search the docker container images in registry server from the terminal, Let’s assume I want to search latest debian Docker Images.

[root@docker ~]# docker search debian

Output of above command would be something like below

**Example:4 Downloading Docker Container Images**

With the help of ‘docker pull’ command we can download docker container images from its registry server or repository.

Syntax : # docker pull <Name-of-Image>

Docker pull command always download latest version of image, though we can specify the particular of image. Let’s assume I want to download latest version of Debian Docker image

[root@docker ~]# docker pull debian

Using default tag: latest

latest: Pulling from library/debian

10a267c67f42: Already exists

Digest: sha256:476959f29a17423a24a17716e058352ff6fbf13d8389e4a561c8ccc758245937

Status: Downloaded newer image for debian:latest

[root@docker ~]#

**Downloading Specific OS version Docker Image**

Let’s assume we want to download Ubuntu:14.04 Docker Image

[root@docker ~]# docker pull ubuntu:14.04

14.04: Pulling from library/ubuntu

cf0a75889057: Already exists

c8de9902faf0: Already exists

a3c0f7711c5e: Already exists

e6391432e12c: Already exists

624ce029a17f: Already exists

Digest: sha256:b2a55128abd84a99436157c2fc759cf0a525c273722460e6f8f9630747dfe7e8

Status: Downloaded newer image for ubuntu:14.04

[root@docker ~]#

**Example:5 List all Downloaded Docker Container Images**

Whenever we download docker images using docker pull command, it save the docker images locally(/var/lib/docker) on your Docker Server. We can list all the downloaded docker images using the ‘**docker images**‘ command,

[root@docker ~]# docker images

REPOSITORY             TAG                 IMAGE ID            CREATED             SIZE

wordpress              latest              ca96afcfa242        2 weeks ago         406 MB

xibosignage/xibo-xmr   release\_1.8.1       223afb5ecffe        2 weeks ago         269 MB

ubuntu                 16.04               ebcd9d4fca80        2 weeks ago         118 MB

ubuntu                 14.04               2ff3b426bbaa        2 weeks ago         188 MB

centos                 7                   8140d0c64310        2 weeks ago         193 MB

mysql                  5.6                 ed7b6c642b9d        3 weeks ago         299 MB

mysql                  5.7                 e799c7f9ae9c        3 weeks ago         407 MB

debian                 latest              3e83c23dba6a        3 weeks ago         124 MB

xibosignage/xibo-cms   latest              9678c5299918        5 weeks ago         511 MB

xibosignage/xibo-cms   release\_1.8.1       c2767fdc7262        5 weeks ago         511 MB

[root@docker ~]#

**Example:6 Launching a container with ‘docker run’ command**

Containers are launched with ‘docker run’ command, let assume I want to launch a container from the debian image.

[root@docker ~]# docker run -it -p 9000:80 --name=debian\_container1 debian

root@9254e01fadad:/#

Above Command launch a container with the name “**debian\_container1**” and we also set pat rule in such a way that if any request comes to 9000 port on docker host then that request will be redirected to debian container on 80 port. If you might have noticed that we directly get the console just after executing the command. Type exit to stop / shutdown the container and if want to get out from the container without exiting then type “**ctrl+p+q**”

**Example:7 Launching a Container in detach mode**

We can launch a container in detach mode using –d option in ‘docker run’ command. Example is shown below

[root@docker ~]# docker run -it -d -p 9001:80 --name=web\_container centos:7

258adf2f71fba82aa04e5f7e9ebba27b8ad87b0439e9bd7d2bcf7e3c40279d54

[root@docker ~]#

**Example:8  Getting Container Console with ‘docker attach’**

In the above example we have provision a container in detach mode, we can have its console using ‘docker attach’ command.

Syntax: # docker attach {container\_name or container\_id}

[root@docker ~]# docker attach web\_container

[root@258adf2f71fb /]#

**Example:9 List Containers with ‘docker ps’ command**

We can list all the running containers using ‘docker ps’ command.

[root@docker ~]# docker ps

To List all the containers including stop /shutdown, use ‘**docker ps -a**‘

**Example:10 Start, Stop, Restart and Kill Containers**

Just like virtual machines we can start, stop and restart docker containers.

Use below command to stop a running container

[root@docker ~]# docker stop db\_container1

db\_container1

[root@docker ~]#

In Place of container we can also use Container ID

Use below command to start a container.

[root@docker ~]# docker start db\_container1

db\_container1

[root@docker ~]#

Use below command to restart a container.

[root@docker ~]# docker restart db\_container1

db\_container1

[root@docker ~]#

Just like Process we can also kill a container, Use below command to kill a container.

[root@docker ~]# docker kill db\_container2

db\_container2

[root@docker ~]#

**Example:11 Delete / Remove a Container**

‘**docker rm**‘ command is used to remove or delete a container. ‘docker rm’ will work only when docker is stopped / shutdown.

Syntax: # docker rm {container\_name\_OR\_container\_id}

[root@docker ~]# docker rm db\_container2

db\_container2

[root@docker ~]#

To remove a running container forcefully, use ‘-f’ option in docker rm command. Example is shown is below

[root@docker ~]# docker rm -f  web\_container2

web\_container2

[root@docker ~]#

**Example: 12 Delete / Remove Docker Container images**

Just like containers we can also delete or remove docker images, ‘**docker rmi**‘ command is used to delete docker container images.

Let’s assume i want to delete a docker image ‘Ubuntu:14.04’

[root@docker ~]# docker rmi ubuntu:14.04

Untagged: ubuntu:14.04

Untagged: ubuntu@sha256:b2a55128abd84a99436157c2fc759cf0a525c273722460e6f8f9630747dfe7e8

Deleted: sha256:2ff3b426bbaafba63cae165e8f6a4955a24a53cdf6d25cce00353e97cda3df71

[root@docker ~]#

In above command in place of image name we also use image id

[root@docker ~]# docker rmi 2ff3b426bbaa

**Example:13  Save and Load Docker Container Image to/from a tar file**

Let’s assume we have done some changes in debian image and we want to export it as tar file, this can be done with ‘**docker save**‘ command, example is shown below

[root@docker ~]# docker save debian -o mydebian.tar

[root@docker ~]# ls -l mydebian.tar

-rw-------. 1 root root 129394176 Jun  1 23:52 mydebian.tar

[root@docker ~]#

**Load Docker Image from tar file.**

[root@docker ~]# docker load -i mydebian.tar

Loaded image: debian:latest

[root@docker ~]#

Note: These commands become useful when we want to transfer docker image from one Docker Host to another.

**Example:14 Export and Import Container to / from tar archive**

Container can be exported to a tar file using the ‘docker export’ command. Syntax is shown below

# docker export container\_name\_or\_id

Let’s suppose I want to export web\_container to a tar file.

[root@docker ~]# docker export web\_container -o  web\_container.tar

[root@docker ~]#

Use ‘**docker import**‘ command to import container from a tar file, examples is shown below.

[root@docker ~]# docker import web\_container.tar

sha256:254b1bdba79775191d33d221eb547e9caca161363639273103193919db71778d

[root@docker ~]#

**Example:15 Display History of Docker Image**

History of Docker image here means what commands are being executed while building docker images, we can list these commands using ‘**docker history**‘

Syntax : # docker history {Image\_Name\_OR\_Image\_id}

[root@docker ~]# docker history ubuntu:16.04

IMAGE               CREATED             CREATED BY                                      SIZE                COMMENT

ebcd9d4fca80        2 weeks ago         /bin/sh -c #(nop)  CMD ["/bin/bash"]            0 B

<missing>           2 weeks ago         /bin/sh -c mkdir -p /run/systemd && echo '...   7 B

<missing>           2 weeks ago         /bin/sh -c sed -i 's/^#\s\*\(deb.\*universe\...   2.76 kB

<missing>           2 weeks ago         /bin/sh -c rm -rf /var/lib/apt/lists/\*          0 B

<missing>           2 weeks ago         /bin/sh -c set -xe   && echo '#!/bin/sh' >...   745 B

<missing>           2 weeks ago         /bin/sh -c #(nop) ADD file:d14b493577228a4...   118 MB

[root@docker ~]#

**Example:16 Fetching Logs from the Container**

We can fetch the logs from the containers without login into it, Use ‘docker logs’ command

Syntax : docker logs {container\_name\_or\_container\_id}

[root@docker ~]# docker logs xibodocker\_cms-web\_1

To fetch the live logs use ‘-f’ option in docker logs command

[root@docker ~]# docker logs -f  xibodocker\_cms-web\_1

**Example: 17 Containers’ resource usage statistics**

To Display CPU, memory, and network I/O usage of all the containers use ‘**docker stats**‘

[root@docker ~]# docker stats

Above command will show live streaming of resource usage statistics of all the Containers.

**Resource usage statistics without live streaming**

[root@docker ~]# docker stats --no-stream

We can display the running processes of a container with ‘**docker top**‘ command.

Syntax: # docker top {Container\_Name\_OR\_ID}

[root@docker ~]# docker top xibodocker\_cms-web\_1

**Example:18 Display Container IP address**

Container low-level information is displayed with ‘**docker inspect**‘ command. We can fetch the ip address of a container from command output. Example is shown below

[root@docker ~]# docker inspect web\_container | grep IPAddress | cut -d '"' -f 4

172.17.0.3

172.17.0.3

[root@docker ~]#

or

[root@docker ~]# docker inspect -f "{{.NetworkSettings.IPAddress}}" web\_container

**Example:19. Build Docker Container Images with Docker file**

With ‘**docker build**‘ command we can build our own docker image with DockerFile.

[root@docker ~]# docker build -t mycentos:web1 .

For More detail on docker file please refer the below

[**How to build Docker Container Images with Dockerfile**](https://www.linuxtechi.com/build-docker-container-images-with-dockerfile/)

**Example: 20 Set Tag / Name to Docker Image**

‘docker tag’ command is used to set tag or name of docker image into a repository.

Syntax : # docker tag source\_image{:tag} target\_image{:tag}

Let’s suppose I want set tag of source image ‘centos:7’ as ‘MyCentOS7:v1’

[root@docker ~]# docker tag centos:7 mycentos7:v1

[root@docker ~]#

Let’s verify the new tag name of docker image

That’s all from this article. I hope these examples might be useful for you to understand docker commands. Please feel free to share your feedback and comments.